Quantification Criteria for Optimization of Modules in OO Design

Pradeep Bhatia
Department of Computer Science & Engineering
G. J. University, Hisar -125 001(INDIA)
E-mail : pk_bhatia2002@yahoo.com

Yogesh Singh
School of Information Technology,
G.G.S. Inderprastha University, Delhi (INDIA)

Abstract
Minimal coupling and maximal cohesion is one of the ways to reduce effort to understand and maintain software systems. When OO paradigm brought a new design philosophy, then encapsulation mechanisms helped us to achieve that desideration. However, after a decade when this paradigm has emerged as the dominant one, coupling and cohesion only do not seem to be the dominant driving forces when it comes to measurement of modularization. This paper introduces a new measure module weakness that helps us to decide quantification of criteria for optimization of modules in OO design.

Keywords: Cohesion, Coupling, Classes, Modular system, Coupling category, Affinity-rating scheme, Similarity and dissimilarity between classes.

1. Introduction

Modularity is an essential aspect in all engineering domains. It is an internal quality attribute influencing the external software quality characteristics. Modularity should be applied at all levels of abstractions ranging from requirement specification to executable code level. Modularity introduced following fundamental engineering concepts:

- To test system in partial fashion.
- To handle complexity of a large system.
- To repair defective parts of a system without interfacing with other parts.
- To design and develop different parts of same system by different people.
- To control defect propagation.
- To reuse existing parts in different contexts.

In this paper, we will analyze the design modularity of OO systems. A module is as set of classes and the coupling among classes within same module(intra-module class coupling) is called *module cohesion* and coupling among classes belonging to different modules (inter-module class coupling) is called *module coupling*. Figure 1 shows the representation of modular system.

2. Quantitative Approach

In this section, we will focus on some quantitative criteria that on conjunction help us to introduce a new quantitative criterion for best modularity.

2.1 Criteria I -The maximum number of classes in a module : Module dimension

Ivar Jacolism mentions 1 to 5 classes as good size for a module; although he admits having used as much as 17. Bertand Myer advocates that software systems should be divided into modules with 5 to 40 classes per module. With this in both above cases uniformity in module dimension should be maintained. Further, this will focus the result in ill-defined modularization. Uniformity is measured with dispersion statistics RMD(Relation Module Dispersion) and is defined as follows[3]:

$$RMD = \frac{CMmax - CMmin}{AMM}$$

where CMmax :Maximum number of classes in current modular system.

CMmin : Maximum number of classes in current modular system.

AMM(Average Modular Modularization) is defined as:

$$AMM = \frac{NC}{NM}$$

RMD=0 when CMmax = CMmin or all modules have the same number of classes.

2.2 Criteria II –The maximum number of modules in modular system: Number of Modules

This can be measured based upon desideration:

"*Maximum module cohesion and minimum module coupling*"

This can be expressed by the measurement ICD(Intra-module Coupling Density) defined as follows[3]:

$$ICD = \frac{CIin}{CIout + CIin}$$

where CIin : Number of coupling between classes within module.

CIout: Number of coupling between classes belonging to distinct modules.

Coheson Level based upon ICD

ICD	Cohesion	Remarks
Highest	Maximum	All the classes resides in same modules(No modularization)
Lowest	Minimum	One class per module(Not an ideal modularization)

Table :1

Thus, we are only concerned with NC-2 different number of modules for NC number of classes, corresponding to NM ∈ [2, NC-1].

2.3 Criteria III - Distribution of classes amongst the modules.

When number of modules is fixed we get to the problem of finding the optimal grouping of classes. This problem can be solved with cluster analysis. Cluster analysis is a set of techniques concerned with the classification of similar items into groups. The objective of clustering techniques is the grouping of items in such a way that the relations between items in the same group are stronger than the relations between items of different groups. Classes that are strongly coupled should belong to same module, while loosely coupled classes should be placed in distinct modules. Figure 1 shows effect of similarity between classes on module cohesion and module coupling.

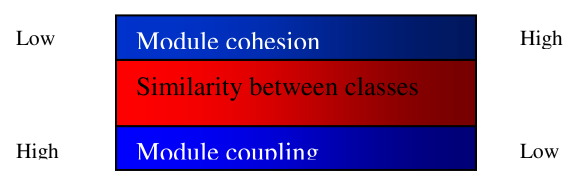

Figure 1 Pattern based similarity between classes

We have following five affinity-rating schemes and seven coupling methods, and each rating scheme includes 12 coupling categories as follows[4]:

Seven Coupling Methods

Sr. No.	Scheme	Definition
1.	Unweighted Binary	$A_{UB}(i, j) = \sum_{cc=DI}^{MR} C_{cc}(i, j)$
2.	Weighted Binary	$A_{WB}(i, j) = \sum_{cc=DI}^{MR} \alpha_{cc} C_{cc}(i, j)$
3.	Unweighted Additive	$A_{UA}(i, j) = \sum_{cc=DI}^{MR} N_{cc}(i, j)$
4.	Weighted Additive	$A_{WA}(i, j) = \sum_{cc=DI}^{MR} \alpha_{cc} N_{cc}(i, j)$
5.	Unweighted Multiplicative	$A_{UM}(i, j) = \prod_{cc=DI}^{MR} N_{cc}(i, j)$

Table : 2

Five-rating Schemes

Sr. No.	Method Name	The distance between two groups is
1.	Single linkage (nearest neighbor)	….the distance between their closest
2.	Complete linkage (furthest neighbor)	…. the distance between their most remote pair of items (opposite of single linkage).
3.	Between-groups linkage (group average)	…. the average of the distances between all pairs of individuals in the two groups.
4.	Between-groups linkage (group average)	….the average of the distances between all pairs of cases in the cluster that would result if they were combined.
5.	Within-groups linkage	….the average of the distances between all pairs of cases in the cluster that would result if they were combined.
6.	Centroid	…. the distance between the group centroids, that means, for all of the items.
7.	Median	…. similar to the centroid but it weights equally the two groups to combine, in the calculation of the centroid.

Table : 3

where:

α_{cc}-positive non-null weight associated with a given CC coupling category,

$CC_{cc}(i,j)$ - predicate with values 1 or 0 sated whether classes i and j are coupled by at least one CC type of coupling or not;

$N_{cc}(i,j)$ - number of instances of CC coupling type, between classes i and j

CC – Coupling Categories viz Direct Inheritance(DI), Class Parameter(CP), Attribute Type(AT), Employed Attribute(EA), Parameter in Operation(PO), Parameter in Message(PM), Parameter in Call(PC), Return in Operation(RO), Return in Message(RM), Return in Call(RC), Local Attribute in Operation(LAO), Message Recipient(MR).

Thus, we have 7x5=35 affinity matrices A_{ij}(i=1 to 7,j=1 to 5) corresponding to seven coupling methods and five rating schemes as follows:

7x5 =35 Affinity square matrices each with size NC x NC

Affinity rating scheme → / Coupling Methods ↓	1	2	3	4	5
I	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
II	A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
III	A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
IV	A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
V	A_{51}	A_{52}	A_{53}	A_{54}	A_{55}
VI	A_{61}	A_{62}	A_{63}	A_{64}	A_{65}
VII	A_{71}	A_{72}	A_{73}	A_{74}	A_{75}

Table : 4

From the affinity values we calculate the dissimilarity D(p,q) among classes using following transformation:

$$D(p,q)=\begin{cases} \dfrac{1}{1+A(p,q)} \forall p \neq q \\ 0 \ \forall p = q \end{cases}$$

Thus, we have 7x5=35 dissimilarity matrices such that each D(p,q)$\in$ [0,1] and each square symmetrical matrix with zeros on the main diagonal; p,q= 1 to NC.

3. Proposed Model

The number of classes, number of modules in a modular system, and the way in which the classes are distributed among the modules is very important to find out the strength of a modular system. It is computed with the help of metric module weakness. If the value of module weakness is less, it will have an adverse effect on testing and maintenance [2]. Module weakness is one of the important metric able to identify OO designs that are likely to error-prone or difficult to maintain or test.

Some quantitative criterion is required to achieve the optimized modularization configuration. We have proposed new model to measure module weakness keeping in view criteria I, II, and III simultaneously, depicted in section 2.

Proposed model to measure modularity of a modular system:

$$WM = \frac{NC}{NM} + CI_{out} - CI_{in} \qquad \text{………… (1)}$$

This model based upon the principle:

"***For a fixed number of modules, distribution of classes such that maximizes the module cohesion and minimizes the module coupling so that module weakness(WM) is minimum***".

For overall i.e for all modules, module weakness is minimized for best of the best modularization.

With seven clustering methods and five affinity-rating schemes, we obtain 7x5= 35 modularization solutions for fixed number of modules. We thus have to find best solution out of (NC-2)*35 modularizations for given NC classes.

Process involves following three steps:

Step 1: Finding the module weakness corresponding to each 7x5=35 dissimilarity matrix using (1). [Appendix A].

Step 2: Finding the best solution for a fixed number of modules among 35 modularizations.

Step 3: Comparing the best solution found for each of the NC-2 modules decomposition alternatives to derive the best of the best.

Figure 2 shows the architecture of the process.

After Step 1

Module weakness corresponding to each dissimilarity matrix

Number of Modules	Module Weaknesses
2	WM(i,j), i =1 to7, j=1 to 5
3	WM(i,j), i =1 to7, j=1 to 5
4	WM(i,j), i =1 to7, j=1 to 5
5	WM(i,j), i =1 to7, j=1 to 5
--	-------------------------
--	-------------------------
NC-1	WM(i,j), i =1,7; j=1,5

Table : 5

After Step 2

The best solution for a fixed number of modules

Number of Modules	Module Weaknesses
2	WM(2) = min(WM(i,j), i =1 to7,j=1 to 5)
3	WM(3) = min(WM(i,j), i=1 to7,j= 1 to 5)
4	WM(4) = min(WM(i,j), i =1 to7,j=1 to 5)
5	WM(5) = min(WM(i,j), i=1 to7,j= 1 to 5)
--	--------------------------------
--	--------------------------------
NC-1	WM(NC-1) = min(WM(i,j), i=1 to7,j=1 to 5)

Table : 6

After Step 3

The best solution is then, among the set of NC-2 solutions derived in the pervious step:

WM = min(WM(k), k=2,NC-1)

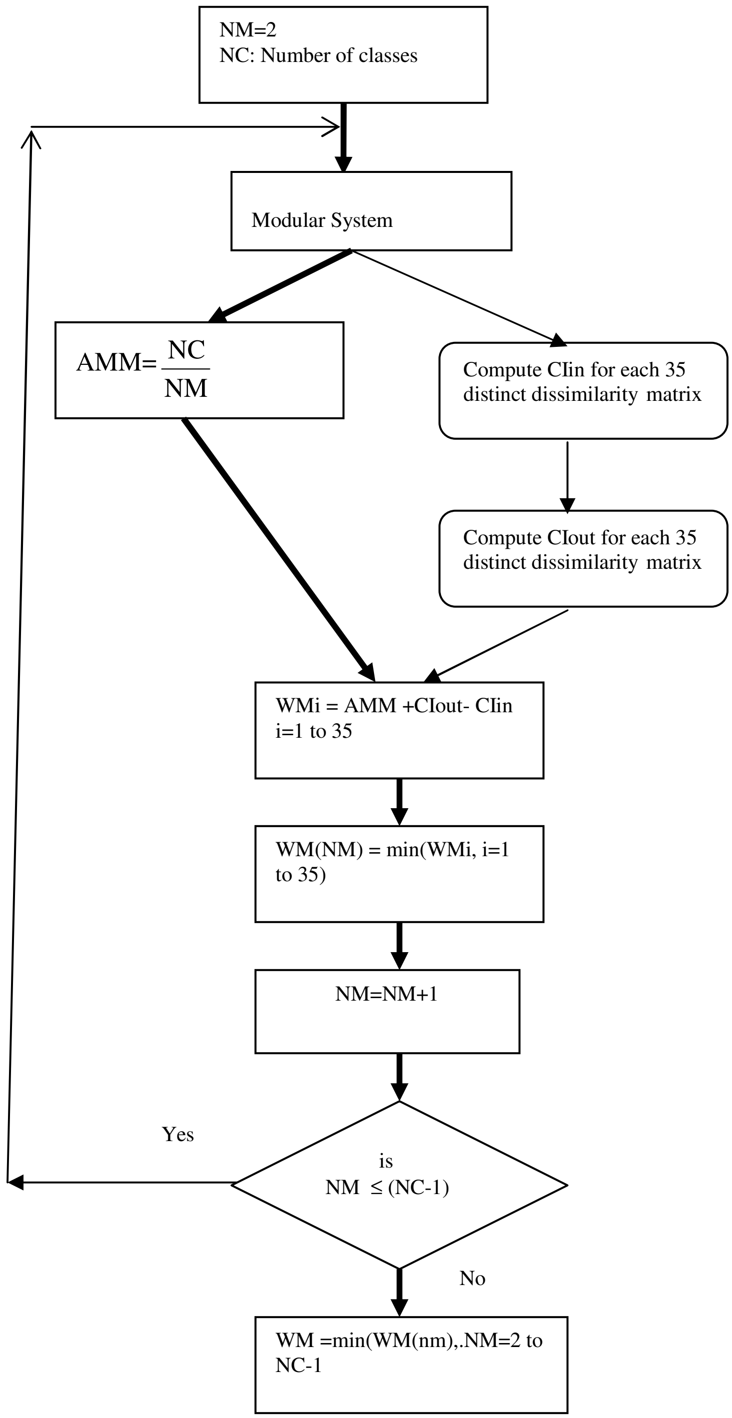

Figure 2: Architecture of the Process

4. Conclusions

The present study introduces an important framework for computing module weakness of an OO design. Once we have measured module weakness, it can be used for best modularization solution. It is a practical framework since all the dependent parameters can be computed for a given modular system. On the basis of measurement of module weakness we will be able to characterize its quality better, assess it with regard to other systems and predict its product quality, e.g. maintaining effort, and error-proneness. This approach presented here can be used for initial design as well as for re-engineering of object-oriented system to identifying ill-defined modularizations and proposing alternative ones. Modularization criterion is concerned besides driving forces coupling and cohesion, number of classes and number of modules.

References

1. Pradeep Bhatia, Yogesh Singh, H.L.Verma, "A Model for Weakness Measurement of Modular System", Ultra Scientist of Physical Sciences, Vol. 15, No. 1, pp. 71-78, April 2003.
2. Yogesh Singh, Pradeep Bhatia, "Module Weakness: A New Measure", ACM SIGSOFT Software Engineering Notes, Vol. 23, No. 04, Page 82, July 1988.
3. Fernando Brito e Abreu, Miqual Goulao, "Coupling and Cohesion as Modularization Drivers: Are we being over-persuaded"?, http://ctp.di.fct.unl.pt/QUASAR/Resources/ papers /Reengineering%20of%20Modularization/csmr.
4. Fernando Brito e Abreu, Miqual Goulao, "A Merit Factor Driven Approach to the Modularization of Object-Oriented Systems", Published in L'Objet vol. 7, no. 4, 2001, Hermes Publisher, France.
5. Conte, S.K., Dunsmore, H.E., Shen, V.Y., *Software Engineering Metrics and Models*, The Benjamu/Commings Pub. C. Inc. California, USA(1986).
6. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw Hill Publishing, 1997.
7. Fenton N., *Software Metrics: A Rigorous Approach*, Chapman and Hill, 1991.
8. Jacobson I., Christerson M., Jonsson P., and Overgaard G., *Object-Oriented Software Engineering-A Use Case Driven Approach*, Addison-Wesley/ACM Press, 1992.
9. Rumbaugh J., Blaha M., Premerlani W., Body F. Lorensen W., *Object-Oriented Modeling and Design*, Englewood Cliffs NJ, EUA: Prentice Hall, 1991.
10. Briand L., Daly J.W., Wust J. K., "A Unified Framework for Coupling Measurement in Object-Oriented Systems", IEEE Transactions on Software Engineering, Vol. 25, No. 1, 1999.
11. Ghezzi C., Jazayeri M., Mandrioli D., *Fundamentals of Software Engineering*, Prentice Hall, 1991.
12. Hutchens D. H., Basili V.R., " System Structure Analysis : Clustering with Data Bindings", IEEE Transactions on Software Engineering, Vol. 11, No. 08, pp 749-757, 1985.
13. Sommervilla I., *Software Engineering*, 6TH ed. Addison Wesley, Logman, 2000.
14. Tonella P., Concept Analysis of Module Restructuring, IEEE Transactions on Software Engineering, Vol. 27, No. 04, pp 351-363, April 2001.

Appendix A : To compute Module Weakness for each dissimilarity matrix for fixed number of modules say NM.

```
for(i= 1, j<= 7, j++)
{
   for (j=1,j<=5,j++)
   {
    CIin=0; CIout=0;
    /* Computation of CIout & CIin----------*/
      for(p =1, p<=NC, p++)
         {
         for(q =l, q<=p, q++)
              {
                if (Dij(p,q) < > 0)
                  CIout = CIout +1;
                else
                  CIin   =  CIin  +1;
              }
        }
 /*------------------------------------*/
```

```
       /*Computation of Module Weakness*/
       WMij(NM) = NC/NM + CIout –CIin;
      }
}
```
